

CDS 130 Computing for Scientists

Mid-term exam review

Topics:

1. Binary Number Representation
 - 1) Bases
 - 2) Binary to decimal number conversion
 - 3) Decimal to binary number conversion (template method)
 - 4) Decimal to hex number conversion
 - 5) Binary to hex number conversion
2. Bit Patterns
 - 1) number of bit patterns for N bits: 2^N
 - 3) unique combinations of 0s and 1s in N bits: 2^N
 - 2) the largest integer number represented by N bits: 2^N-1
3. Binary Arithmetic
 - 1) Binary number addition
 - 2) Binary number subtraction
 - 3) Binary number multiplication
4. Binary representation of negative numbers
 - 1) sign-and-magnitude representation and its range
 - 2) two's complement representation and its range
5. Encoding
 - 1) ASCII code
 - 2) Encode words
 - 3) Interpret ASCII codes
6. Computing Limitation
 - 1) Overflow
 - 2) Data range represented by N bits
7. Data storage
 - 1) bits, bytes, words, and doubles
 - 2) Kilobytes, Megabytes, and Gigabytes.
 - 3) Difference between memory and data storage
8. Logic gates
 - 1) AND, OR, NAND, NOR, ~~XOR~~, ~~XNOR~~, NOT gates
 - 2) Truth tables for the logic gates
 - 3) Building a logic gate
 - 4) logic circuits

MATLAB

9. Variables

- 1) Scalars, vectors, matrices, strings
- 2) Assignment

10. Create vectors : Row vector and Column vector

method 1: $A(5) = 1;$

method 2: $A = [2,3,4,5];$ or $A=[2; 3; 4; 5];$

For row vectors, you can use either commas or spaces to separate the elements

method 3: $A = 2:-1:-6;$ (colon notation).

If the increment is 1, the short-handed notation is $A = 2:5;$ (i.e., $A=[2, 3, 4, 5]$)

method 4: form a new vector from previously defined vectors

```
>> A = [1, 2, 3];
```

```
>> B = [3, 4, 5];
```

```
>> C = [A, B]
```

method 5: iterations

```
>> A(1) = 0;
```

```
>> for I = 2:6
```

```
    A(I) = I
```

```
end
```

% this creates a new vector $[0, 2, 3, 4, 5, 6]$, and it does so by appending new pieces onto an existing vector, one piece at a time.

* Not tested: `linspace()`, `zeros()` and `ones()`.

11. Vectors in Matrix

1) row vector: $A(5)$ means $A(1, 5)$

2) column vector $A(5,1)$

3) Transpose operator: $A(5)'$ is now a column vector

12. Address vector elements

1) meaning of $A(1,5)$ and $A(5)$

2) meaning of $A(5,1)$

3) $A(1:3)$ is a new vector

4) $A(0.2)$ is invalid

13. Vector operations

1) Vector added by a scalar

```
>> A = [1, 2, 3];
```

```
>> B = A + 3.0
```

```
B =
```

```
4 5 6
```

3) Vector multiplied by a scalar

```
>> A = [1, 2, 3];
```

```
>> B = A * 3.0;
```

```
B =
```

```
3 6 9
```

2) Add two vectors

```
>> A = [1, 2, 3]
```

```
>> B = [3, 4, 5]
>> C = A+B
C =
    [4, 6, 8]
```

4) Vector summation (sum of all vector elements)

```
>> C = [4, 6, 8];
>> sum(C)
ans = 20
```

5) Element-by-element operation (Attach . to the first vector) (extra credit)

```
>> C = A. * B
C =
    [3, 8, 15]

>> C = [1 1 1]. / A;
C =
    1    0.5  0.33333
```

14. Create Matrices

```
method 1: A(4,5) = 0.0;
method 2: A = [1, 2, 3; 3, 4, 5];
method 3: >> B = [1:4];
>> C = [4:7];
>> A = [B; C];
```

15. Address matrix elements

```
(1) A(5, 4)
(2) A(1:2, 2,3)
(3) A(:, 2:3)
(4) A( : , : )
```

16. Matrix Operations

(1) matrix added by a scalar

```
>> A=[1, 2, 3; 3, 4, 5];
>> B = A + 2.0;
B =
     3     4     5
     5     6     7
```

(2) matrix multiplied by a scalar

(3) matrix addition

(4) element-by-element operation (extra credit)

```
>> A = [1, 2, 3; 4, 5, 6];
>> A.*A
ans =
     1     4     9
    16    25    36
```

```
(5) Sum of matrix
    >> sum(A)
    ans =
        14     77
```

17. Iterations

```
(1) syntax:
    for i=1:10
        do something
    end
```

Example 1:

```
A = [1:0.2:1.5; 0.5:-0.1: 0.3]
for myVar = 0.2:0.5:1.3
    A = A * myVar;
end
```

Example 2:

Considering the following iteration code, what is A(12)?

```
A(9)=13;
for i=[10:12] ;
    A(i)=A(i-1)+37;
end
```